## AMENDMENTS TO THE CLAIMS

1. (Currently amended) A network interface device, comprising:

receive logic, which is coupled to receive from a network a sequence of data packets, each packet comprising respective header data;

a protocol processor, coupled to read and process the header data so as to identify a group of the received packets that contain respective fragments of a data frame, the fragments having a fragment order within the data frame; and

host interface logic, which is coupled to a host memory accessible by a host processor, and is controlled by the protocol processor so ~~as to~~ that irrespective of whether the sequence in which the packets are received coincides with the fragment order, the host interface logic allocates space for the data frame in the host memory responsively to the header data of the packet received first in the sequence among the packets in the group, and ~~to~~ reassembles the fragments of the data frame in the fragment order in the space allocated in the host memory.

2-3. (Canceled)

4. (Original) A device according to claim 1, wherein the protocol processor is arranged to control the host interface logic so as to write each of the fragments to a respective location within the allocated space in the host memory responsive to a fragment offset parameter in the header data of each of the packets.

5. (Original) A device according to claim 1, wherein the protocol processor is arranged to control the host interface logic so as to allocate the space for the data frame responsive to the packet that is received first in the sequence among the packets in the group.

6. (Original) A device according to claim 5, wherein the protocol processor is arranged to determine a size of the space to allocate responsive to a frame length indication in the header data of the packet received first in the sequence.

7. (Original) A device according to claim 6, wherein the frame length indication comprises one or more fields in the header data indicating an exact length of the data frame.

8. (Original) A device according to claim 6, wherein the frame length indication comprises one or more fields in the header data indicating an upper bound on a length of the data frame.

9. (Original) A device according to claim 1, wherein the protocol processor is arranged to monitor a time required to receive all of the packets in the group, and to control the host interface logic so as to release the space allocated for the data frame if the time exceeds a predetermined limit without all of the fragments in the data frame having been reassembled.

10. (Original) A device according to claim 9, wherein when the time exceeds the predetermined limit without all of the fragments in the data frame having been reassembled, the protocol processor is arranged to return a message over the network to a source of the packets indicating that the data frame was not received.

11. (Original) A device according to claim 1, wherein the group of the received packets is one of a plurality of different groups, the packets in the different groups containing the fragments of different, respective data frames, and wherein the protocol processor is arranged to identify the different groups and to control the host interface logic so as to simultaneously reassemble the fragments of the different data frames in respectively-allocated spaces in the host memory.

12. (Original) A device according to claim 1, wherein the packets are transmitted over the network and received by the receive logic in accordance with one or more communication protocols, whereby the header data in the packets comprise protocol information, and wherein the protocol processor is arranged to identify the group and to control the host interface logic responsive to the protocol information.

13. (Original) A device according to claim 12, wherein the one or more communication protocols comprise a plurality of different communication protocols, and wherein the protocol processor is arranged to select the group of packets for reassembly depending on which of the communication protocols was used in transmitting the packets.

14. (Original) A device according to claim 12, wherein the one or more protocols comprise a network layer protocol.

15. (Original) A device according to claim 14, wherein the network layer protocol comprises an Internet Protocol (IP).

16. (Original) A device according to claim 12, wherein the one or more protocols comprise a transport layer protocol.

17. (Original) A device according to claim 16, wherein the transport layer protocol comprises a Transport Control Protocol (TCP).

18. (Original) A device according to claim 16, wherein the transport layer protocol comprises a User Datagram Protocol (UDP).

19. (Original) A device according to claim 1, wherein the receive logic is coupled to receive the packets over the network from a plurality of different sources, and wherein the protocol processor is arranged to select the group of packets for reassembly dependent on the packets having been received from one or more chosen sources among the plurality of different sources.

20. (Original) A device according to claim 19, wherein the selected sources are chosen responsive to a level of reliability of a connection over the network between the chosen sources and the network interface device.

21. (Original) A device according to claim 1, wherein the protocol processor is arranged to control the host interface logic so as to write the data packets that do not belong to the identified group to the host memory substantially without reassembly processing thereof by the network interface device.

22. (Original) A device according to claim 1, wherein the host interface logic comprises a direct memory access (DMA) engine, which is arranged to write the fragments to the host memory substantially without involvement of the host processor.

23. (Original) A device according to claim 22, wherein the host interface logic is coupled to notify the host processor that the fragments of the data frame have been reassembled in the host memory only after all of the fragments have been reassembled.

24. (Original) A device according to claim 23, wherein the protocol processor is arranged to determine a total length of the data frame responsive to the header data of at least one of the packets in the group, and to count a quantity of the data in the fragments reassembled in the data frame, and to determine that all of the fragments have been reassembled by comparing the total length to the quantity of the data reassembled.

25. (Original) A device according to claim 23, wherein the protocol processor is further arranged to control the host interface logic to write a frame header to the allocated space in the host memory, indicating to the host processor a total length of the data frame.

26. (Original) A device according to claim 1, wherein the receive logic, protocol processor and host interface logic are contained together in a single integrated circuit chip, which is separate from the host processor and host memory.

27. (Currently amended) A method for interfacing a host processor to a network, comprising:

receiving a sequence of data packets from the network at a network adapter, each packet comprising respective header data;

processing the header data in the adapter so as to identify a group of the received packets that contain respective fragments of a data frame, the fragments having a fragment order within the data frame;

allocating space for the data frame in a host memory accessible by the host processor, responsively to the header data of the packet received first in the sequence among the packets in the group, regardless of whether the packet received first in the sequence is first in the fragment order;

writing the fragments of the data frame from the network adapter to the space allocated in the host memory so that the fragments are reassembled in the space in the fragment order irrespective of whether the sequence in which the packets are received coincides with the fragment order; and

notifying the host processor when all of the fragments have been reassembled.

28-29. (Canceled)

30. (Original) A method according to claim 27, wherein writing the fragments comprises writing each of the fragments to a respective location within the allocated space in the host memory responsive to a fragment offset parameter in the header data of each of the packets.

31. (Original) A method according to claim 27, wherein allocating the space comprises allocating the space for the data frame responsive to the packet that is received first in the sequence among the packets in the group.

32. (Original) A method according to claim 31, wherein allocating the space comprises determining a size of the space to allocate responsive to a frame length indication in the header data of the packet received first in the sequence.

33. (Original) A method according to claim 32, wherein the frame length indication comprises one or more fields in the header data indicating an exact length of the data frame.

34. (Original) A method according to claim 32, wherein the frame length indication comprises one or more fields in the header data indicating an upper bound on a length of the data frame.

35. (Original) A method according to claim 27, and comprising monitoring a time required to receive all of the packets in the group, and releasing the space allocated for the data frame if the time exceeds a predetermined limit without all of the fragments in the data frame having been reassembled.

36. (Original) A method according to claim 35, and comprising, when the time exceeds the predetermined limit without all of the fragments in the data frame having been reassembled, returning a message over the network to a source of the packets indicating that the data frame was not received.

37. (Original) A method according to claim 27, wherein the group of the received packets is one of a plurality of different groups, the packets in the different groups containing the fragments of different, respective data frames, and wherein processing the header data comprises identifying the different groups so as to

simultaneously reassemble the fragments of the different data frames in respectively-allocated spaces in the host memory.

38. (Original) A method according to claim 27, wherein the packets are transmitted over the network and received by the receive logic in accordance with one or more communication protocols, whereby the header data in the packets comprise protocol information, and wherein processing the header data comprises identifying the group and preparing the fragments for writing responsive to the protocol information.

39. (Original) A method according to claim 38, wherein the one or more communication protocols comprise a plurality of different communication protocols, and wherein identifying the group comprises selecting the group of packets for reassembly depending on which of the communication protocols was used in transmitting the packets.

40. (Original) A method according to claim 38, wherein the one or more protocols comprise a network layer protocol.

41. (Original) A method according to claim 40, wherein the network layer protocol comprises an Internet Protocol (IP).

42. (Original) A method according to claim 38, wherein the one or more protocols comprise a transport layer protocol.

43. (Original) A method according to claim 42, wherein the transport layer protocol comprises a Transport Control Protocol (TCP).

44. (Original) A method according to claim 42, wherein the transport layer protocol comprises a User Datagram Protocol (UDP).

45. (Original) A method according to claim 27, wherein receiving the sequence of data packets comprises receiving the packets over the network from a plurality of different sources, and wherein processing the header data comprises selecting the group of packets for reassembly dependent on the packets having been received from one or more chosen sources among the plurality of different sources.

46. (Original) A method according to claim 45, wherein selecting the group of packets comprises choosing the one or more sources responsive to a level of reliability of a connection over the network between the chose sources and the network interface method.

47. (Original) A method according to claim 27, and comprising writing the data packets that do not belong to the identified group to the host memory substantially without reassembly processing thereof by the network adapter.

48. (Original) A method according to claim 27, wherein writing the fragments comprises writing the fragments by direct memory access (DMA), substantially without involvement of the host processor.

49. (Original) A method according to claim 27, wherein processing the header data comprises determining a total length of the data frame responsive to the header data of at least one of the packets in the group, and wherein writing the fragments comprises counting a quantity of the data in the fragments reassembled in the data frame, and comprising determining that all of the fragments have been reassembled by comparing the total length to the quantity of the data reassembled.

50. (Original) A method according to claim 27, wherein processing the header data comprises determining a total length of the data frame responsive to the header data of at least one of the packets in the group, and comprising writing a frame header to the allocated space in the host memory, indicating to the host processor a total length of the data frame.